一种 Credit 调度算法的改进算法

张 颜
(武汉理工大学计算机科学与技术学院 武汉 430070)

摘 要 通过分析 Vmware ESX 和 Xen 中的 CPU 调度算法,发现其调度算法都以分区队列模型为基础,因此提出建立共享队列的模型对 Credit 算法进行改进,然后运用排队论对模型进行理论分析和模拟实验。根据模拟实验结果对改进的调度算法进行性能评估。

关键词 虚拟化,Xen,Credit 调度算法,排队论,共享队列模型
中图法分类号 TP393 **文献标识码** A

Improved Scheduler of Credit

ZHANG Yan
(School of Computer Science and Technology, Wuhan University of Technology, Wuhan 430070, China)

Abstract In this paper, through the analysis of VMware ESX and Xen CPU scheduling algorithm, it is found that scheduling algorithms are based on partition queue model. We proposed to establish a shared queue model to improve the credit algorithm, and then carried on the theoretical analysis and simulation experiments of the model. Based on the simulation results, the performance evaluation of the improved scheduling algorithm was performed.

Keywords Virtualization, Xen, Credit scheduling algorithms, Queueing theory, Sharing queue model

1 前言

虚拟化技术第一次出现是在 20 世纪 60 年代,但是近几年来随着云计算的兴起,又重新成为了热门的研究领域,吸引了越来越多的企业和学术研究者对其进行深入研究。虚拟化技术的核心是虚拟机监控器(Virtual Machine Monitor, VMM),它能在物理硬件之上创建多个能独立运行的虚拟机(Virtual Machine, VM)操作系统,这些虚拟机在同一平台能隔离运行,使得各个虚拟机在共享平台下有独立的资源分配[1]。虚拟化研究的意义在于:1)通过利用虚拟机监控器去共享底层物理资源提高资源的利用率;2)随着多核和多处理器计算机系统架构的普及,个人电脑的计算能力变得越来越强,而虚拟化技术的发展相较于快速发展的个人电脑的计算能力则要缓慢得多,因此底层硬件的迅速发展给虚拟机中的中央处理器(Central Processing Unit, CPU)调度算法带来了新的技术挑战。如何将物理中央处理器(Physical Central Processing Unit, PCPU)公平效率地分配给虚拟中央处理器(Virtual Central Processing Unit, VCPU)成为当前虚拟化技术领域的一个非常重要的研究热点。

本文主要研究 VMM 多核处理器中 CPU 调度算法的队列模型设计。由于排队论是当前计算机人员的重要工具之一,通过排队论来分析改进 CPU 调度算法,并对改进后的性能进行评估比较。

本文第 2 节介绍并分析了 Xen 和 Vmware ESX 的 CPU 调度算法策略;第 3 节提出了一种利用共享队列模型来替代传统虚拟机系统中的分区队列模型(Partition Queue Model)的改进策略;第 4 节分别对共享队列模型和分区队列模型进行建模;第 5 节通过仿真实验的结果阐述改进前和改进后系统的性能优劣;最后总结全文。

2 Vmware ESX 和 Xen 中的 CPU 调度机制

当今,Vmware 和 Xen 都提供了运行在对称多处理器架构上的高性能虚拟化软件为客户提供服务器整合的功能,服务器整合是指可以把多个应用服务或操作系统整合到一台或多台物理主机上的多个虚拟机中,并同时维持其正常运行。目前,Vmware EXS 服务器和 Xen 都支持创建多台虚拟机服务,他们允许虚拟机去配置多个 VCPU 模拟多线程在 PCPU 上并发执行。Xen 和 Vmware 的 CPU 调度指如何将 PCPU 分配给虚拟机的 VCPU。详细的 Vmware EXS 和 Xen 调度机制将在下一段落进行描述和分析。

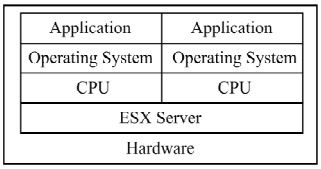

图 1 ESX 服务器的架构图

ESX 服务器能提供高性能服务的主要原因是有 Vmware server 这样的主机虚拟化产品,使得 VMM 能够直接控制系统的硬件资源,而不用先请求宿主系统。ESX 服务器的架构如图 1 所示。VCPU 调度是指 VMM 直接控制 PCPU 的分配,而不是仅仅作为宿主计算机的一个线程来执行。Vmware ESX server 使用的是 VCPU 联合调度算法,联合调度算法是指虚拟机中的 VCPU 同时并发运行在 PCPU 上。EXS 中的联合调度算法能保证虚拟机的正确性和高性能,其缺点是可

张 颜(1992—),男,硕士生,主要研究方向为虚拟化技术。

能会因为 PCPU 等待 VCPU 的调度而被闲置，导致该算法效率较低。

Xen 4.0 版本提供两种调度策略：SEDF(Simple Earliest Deadline First)调度算法和 Credit 调度算法。在当前的版本中默认的调度算法是 Credit，SEDF 已经逐步被淘汰。在 Credit 调度算法中，每个 PCPU 管理一个可运行的队列，根据 VCPU 的优先级将其进行排列入队，优先级有两种状态：OVER和 UNDER，OVER 表示 VCPU 已经超过了预先分配的 CPU 资源，UNDER 则表示 VCPU 没有超过预先分配的 CPU 资源。入队的算法是基于先来先服务(First Come First Service，FCFS)的调度策略将 VCPU 插入具有相同优先级的 VCPU 的后面。在 Xen 中的 Credit 调度算法通过管理调度多个物理 CPU，从而能将物理 CPU 公平高效地分配给各个虚拟 CPU，但是由于一个虚拟 CPU 最多只能占用一个物理 CPU 的资源，容易导致处理器因为进程未就绪而空闲的情况，因此在处理延迟敏感的程序或者实时应用程序时效率较低。为了解决这个问题，Govindan 等人提出了一个改进的 CPU 调度算法——communication-aware CPU scheduling，该算法的主要改进点是优先调度通信密集型虚拟机的调度算法[2]。Diego Ongaro 和 Alan L. Cox 在 Credit 算法中为 VCPU 加入了一个新的状态 BOOST，并且优化了消息事件通道。该改进调度算法虽然降低了事件响应延迟，但是仍然无法缓解 I/O 密集型服务中的虚拟机监控器的负载压力[3]。王凯提出了一种基于 Credit 算法的自适应调整虚拟机权重参数的优化调度算法，其原理是将特权服务操作系统的 I/O 处理能力作为虚拟机参数调整的一个重要参数[4]。

以上所有关于 CPU 的调度算法都是依赖于一个特定条件实现的高性能。该条件就是这些调度算法是基于每个物理 CPU 下都有一个虚拟 CPU 的运行队列。由于物理 CPU 中的分区运行队列在实时性要求较高的系统中有响应时间过长的缺点，本文接下来的部分将论述一种共享队列模型(Sharing Queue Model，SQM)来提高 CPU 调度算法的事件响应速度。

3 建模分析

基于以上在 VMM 中的典型调度算法的描述可知，由于每个物理 CPU 下都有一个虚拟 CPU 的运行队列，导致在 Xen 中的 Credit 调度算法对多个物理 CPU 之上的 VCPU 调度没有很好的负载均衡特性。从而推断出，当所有的物理 CPU 共享一个 VCPU 运行队列比利用分区机制的多个运行队列进行负载均衡或许能实现更好的性能。

为了对比分析，将 Xen 中 Credit 调度算法中的分区队列模型与本文提出的一个共享队列模型(Sharing Queue Model，SQM)进行对比，比较两种不同队列机制的性能表现。

假设有 N 个物理 CPU 且虚拟机上共有 M 个 VCPU，那么在 Xen 中 Credit 调度算法的队列模型和本文提出的共享队列调度算法模型如图 2 和图 3 所示。

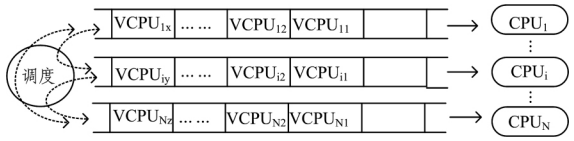

图 2 Credit 调度算法中的分区队列模型

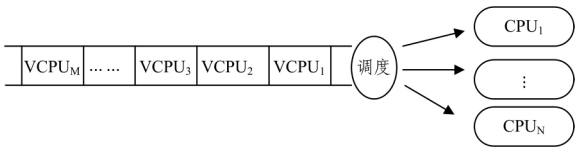

图 3 本文提出的共享队列调度算法模型

虽然该改进调度算法只是在 Credit 算法上进行了一点微调，但以下部分将论证该改进算法对性能的改善有明显的效果。

4 调度算法建模

分区队列模型达到最好的状态就是根据虚拟 CPU 处理快慢的性能差异对 CPU 的处理时间进行合理的分配。在本节中，为了研究更加理想情况下的性能差异，将所有 VCPU 处理性能设置为一样。根据排队论，当 PQM 达到最佳状态时分析分区队列模式(PQM)和共享队列模型(SQM)的性能表现。在测试这个模型的性能指标之前，进行以下的设定：

1)VMM 系统创建的 VCPU 和 VCPU 的排队过程是相互独立的，所有 VCPU 的到达过程符合泊松分布，而且假定到达率为 λ。

2)设定物理 CPU 为虚拟 CPU 服务符合指数分布，且服务率为 μ。

3)在 Xen 的 Credit 调度算法中的 VCPU 分区队列模型(PQM)和共享队列模型(SQM)都符合 FCFS 规则。

4)为了对这两种类型的队列进行比较，对于所有分区运行队列总的到达率和本文提出来的共享队列到达率保持相等。当 CPU 的数量为 N 个，且共享队列的到达率为 λ 时，可以推导出其中一个分区队列的 VCPU 到达率是 λ/N。

基于以上设定，下面将详细分析并讨论分区队列模型和共享队列模型。

4.1 分区队列模型

Xen 中的 Credit 调度算法能在有 N 个 PCPU 的系统上建立 N 个分区队列，队列模型如图 4 所示。

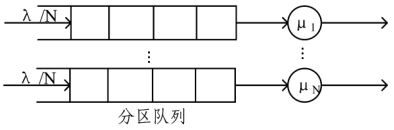

图 4 分区队列模型

因为输入的参数保持一致，所以当使用 PQM 时调度系统里所有队列里的响应时间即可通过计算得出。根据排队论(Queuing Theory)，性能参数响应时间通过式(1)进行计算。

$$T_{q1}=\frac{1/\mu}{1-\rho} \tag{1}$$

在本文中通过 $\rho=\frac{\lambda/N}{\mu}=\lambda/N\mu$ 来表示 CPU 的利用率。

VCPU 的等待时间通过以下的公式来表示：

$$T_{w1}=\frac{\lambda/N}{1-\rho}-\frac{1}{\mu}=\frac{\rho}{\mu(1-\rho)} \tag{2}$$

4.2 共享队列模型

共享队列提出的调度模型如图 5 所示。

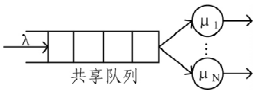

图 5 共享队列模型

同样的响应时间的性能参数通过式(3)进行表示。

$$T_{q2}=\frac{q_2}{\lambda} \tag{3}$$

其中，q_2 通过式(4)进行计算：

$$q_2=N\rho+\frac{(N\rho)^N}{N!}\frac{\eta_0}{(1-\rho)^2} \tag{4}$$

η_0 通过式(5)进行计算：

$$\eta_0=\left[\sum_{k=0}^{N-1}-\rho\frac{(N\rho)^k}{k!}+\frac{(N\rho)^N}{N!}\frac{1}{(1-\rho)}\right]^{-1} \tag{5}$$

CPU 的利用率 ρ 跟分区队列模型的 CPU 利用率计算公式相同：

$$\rho=\frac{\lambda/N}{\mu}=\lambda/N\mu$$

另外，通过式(6)来计算 VCPU 在队列中等待的概率：

$$P[waiting]=\sum_{k=N}^{\infty}\eta_k=\frac{(N\rho)^N}{N!}\frac{\eta_0}{(1-\rho)} \tag{6}$$

当计算出 ρ、μ 和 N 的值，即可计算出响应时间的性能参数。经过计算，得出 $Tq_1 \geqslant Tq_2$。即当两种调度算法中的 VCPU 的数量、VCPU 的到达率、物理 CPU 的数量和 PCPU 的服务速率的值保持一致时，单从响应时间这项性能指标来看，共享队列模型的策略更加高效。

接下来，通过仿真实验来证实上面推导出的结论。

5 仿真实验和结果分析

5.1 实验环境

运行模拟实验的硬件环境如表 1 所列。

表 1 模拟实验的硬件环境

硬件	性能参数
CPU 信息	Intel Core i3 4150
主板信息	华硕 H81M-D R2.0
内存容量	4G
硬盘容量	SATA500G

软件环境信息表如表 2 所列。

表 2 软件环境信息

软件	版本备注
操作系统	MicrosoftWindows 7
Java 运行平台	Java™ SE Development Kit 6 Eclipse SDK
模拟器	Schedsim v1.0

5.2 实验结果分析

在这部分，通过编写 Java 程序实现用不同的队列模型来模拟调度算法，并在模型中输入参数 λ、μ 和 N。给定一个确定增量的 CPU 利用率运行模拟实验，结果如图 6－图 10 所示。

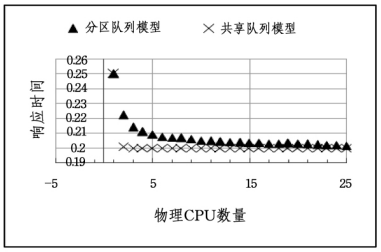

图 6 当 $\lambda/\mu=0.2$ 时物理 CPU 的数量和响应时间对应的关系

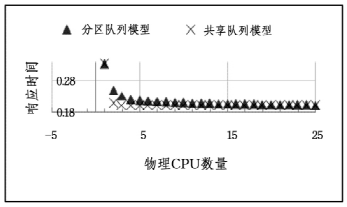

图 7 当 $\lambda/\mu=0.4$ 时物理 CPU 的数量和响应时间对应的关系

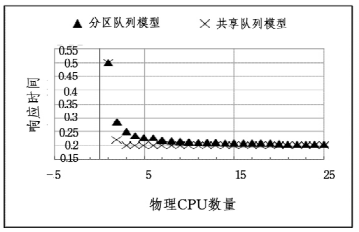

图 8 当 $\lambda/\mu=0.6$ 时物理 CPU 的数量和响应时间对应的关系

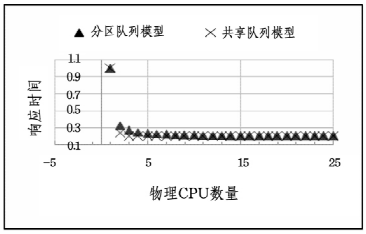

图 9 当 $\lambda/\mu=0.8$ 时物理 CPU 的数量和响应时间对应的关系

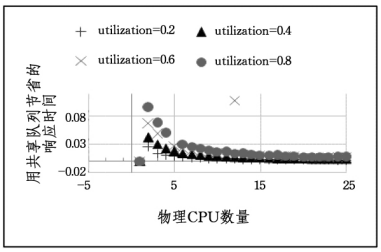

图 10 在 CPU 使用率不同、CPU 个数不同时，使用共享队列节省的响应时间

从图 6－图 10 的模拟结果可知，在其他条件保持一致时，仅从响应时间这项性能指标来看，共享队列模型比分区队列模型能达到更好的性能。而且，随着 CPU 利用率的增加，共享队列模型中的响应时间降低得更加明显(见图 10)。因此能得出结论，随着多核多处理器架构的发展，同时更多的服务器利用 VMM 去整合服务资源导致 CPU 利用率不断地提高。在这种情况下基于共享队列模型将比分区队列模型实现的 CPU 调度算法更高效。

结束语 在本文中，研究了在非单核处理器上 Xen 和 VMWare ESX 中的 CPU 调度算法。Vmware ESX 和 Xen 都利用了分区队列模型来实现调度算法，这表示着每个物理 CPU 都有自己的 VCPU 运行队列。因此，本文通过排队理论进行分析，使用共享队列模型去改进 CPU 调度算法，通过公式推导和模拟程序实现所有物理 CPU 共享一个运行队列，根据实验结果推导出，在响应时间上共享队列模型比分区队列模型具有更好的性能。

在接下来的工作中，将尝试在开源的 Xen 中真正实现共享队列模型 CPU 调度算法，并在不同的工作负载以及不同虚拟机数量的情况下评估传统 CPU 调度算法和本文提出的调度算法哪一个更好。

参 考 文 献

[1] Goldberg R P. Survey of Virtual Machine Research [J]. Computer, 1974, 7(6): 34-45

[2] Govindan S, Nath A R, Das A, et al. Xen and co.: communication-aware CPU scheduling for consolidated xen-based hosting-platforms[C]// VEE. 2007: 126-136

[3] Ongaro D, Cox A L, Rixner S, et al. Scheduling I/O in virtual machine monitors[C]// Virtual Execution Environments. 2008

[4] 王凯，侯紫峰. 自适应调整虚拟机权重参数的调度方法[J]. 计算机研究与发展，2011，48(11)：2094-2102

[5] 暴锡文. 一种云计算环境下基于 Xen 的虚拟机调度机制[J]. 计算机测量与控制，2014，22(10)：3381-3384

[6] 张莹，李华. 一种虚拟机负载均衡调度算法[J]. 河南科技，2015(12)：30-33

[7] 陈锐忠，齐德昱，林伟伟，等. 一种面向非对称多核处理器的虚拟机集成调度算法[J]. 计算机学报，2014，37(7)：1466-1477

[8] 陈锐忠. 非对称多核处理器的若干调度问题研究[D]. 广州：华南理工大学，2013

[9] 常建忠. VCPU 组调度技术的研究与实现[D]. 长沙：国防科学技术大学，2010

[10] 王凯，侯紫峰. Xen 虚拟 CPU 空闲调度算法[J]. 计算机研究与发展，2013，50(11)：2429-2435

[11] 姚文斌，郑兴杰. 一种改进的 SEDF 调度算法[J]. 小型微型计算机系统，2010，31(3)：446-450

(上接第 239 页)

(a) Suize 重建图　　(b) Mobile 重建图　　(c) Foreman 重建图

图 6　3 个视频序列重建图(压缩比为 30)

在视觉效果上，观察图 5 和图 6 可知，Suize，Mobile，Foreman 3 个视频重建图整体视觉良好，在纹理细节部分有点模糊，如 Suize 的头发、Mobile 的日历数字、Foreman 的眼睛部分。

3 个视频序列各有特点：Suize 纹理比较丰富，包含头发、皮肤等细节，运动幅度较小；Mobile 纹理特别复杂，画面中有多个运动物体，运动规迹平缓；Foreman 纹理丰富度一般，画面中人物运动较多且有场景切换。根据 3 种视频序列的纹理复杂度和运动幅度特性分析可知，Surfacelet 变换和 SPIHT 算法编码相结合适用于纹理复杂度较高、运动不剧烈的视频。实际播放完整的视频序列时，视觉效果更加突出。

结束语　本文把视频信号作为特殊的三维信号，提出基于 Surfacelet 变换并结合 SPIHT 算法编码的视频压缩编码方法。SPIHT 算法自动完成嵌入式编码码流的分配，利用 Surfacelet 变换的分解系数在各层间相关和图像能量集中的特性，能获得较好的视频压缩效果。

实验证明，Surfacelet 变换结合 SPIHT 算法的视频压缩编码方法比 3D-DWT 结合 SPIHT 算法的方法实现了更高的 PSNR 和更好的视觉效果，且较适用于纹理复杂度较高、运动幅度较小的视频。

对于空间复杂度较高的视频序列，其图像的中频和高频 Surfacelet 系数成分中不重要系数的节点形成的零树较少，导致 SPIHT 算法对高频系数的压缩性能不如对低频系数的压缩性能，SPIHT 算法编码效率会受到影响。因此可以采取 Surfacelet 变换结合多种算法的方法，对高频系数用自适应编码，对低频系数用 SPIHT 算法编码，以改进视频压缩效果。在实际应用中，还要考虑到算法的运算效率。视频三维编码需要较多的帧存储器处理，会带来较大的延迟，这还有待提高。

参 考 文 献

[1] Chaudhury K N, Unser M. Construction of Hilbert transform pairs of wavelet bases and optimal time-frequency localization [J]. IEEE Transactions on Signal Processing, 2009, 57(9): 3411-3425

[2] Unser M, Blu T. Mathematical properties of the JPEG2000 wavelet filters [J]. IEEE Transactions on Image Processing, 2003, 12(9): 1080-1090

[3] 冯鹏. 高分辨图像处理用抗混叠 Contourlet 变换的若干关键问题研究[D]. 重庆：重庆大学，2007

[4] Zhang J N, Gilling C J, Kreger K S. System and method for filtering frequency encoded imaging signals: US, US6633162[P]. 2003

[5] 江山，尹忠科，陈帆. 基于 Surfacelet 稀疏重构的视频修复[J]. 数据采集与处理，2012，27(4)：444-449

[6] Lu Y M, Do M N. Multidimensional directional filter banks and surfacelets. [J]. IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society, 2007, 16(4): 918-931

[7] Wang H, Pan X. Video compression coding based on the improved 3D-SPIHT[C]// 2010 International Conference on Computer Application and System Modeling (ICCASM). IEEE, 2010: V14-108-V14-111

[8] Tang G, Gu G. LIS-No-Classification Wavelet Image Coding Algorithm Based on Lifting Scheme[C]// International Workshop on Intelligent Systems and Applications. IEEE, 2009: 1-4

[9] Fradj B B, Zaid A O. Scalable video coding using motion-compensated temporal filtering and intra-band wavelet based compression[C]// European Workshop on Visual Information Processing. 2011: 50-55

[10] Lian S. Secure service convergence based on scalable media coding[J]. Telecommunication Systems, 2010, 45(1): 21-35

[11] Jeon B M, Park S W, Park J H. Method and apparatus for encoding/decoding video signal using block prediction information: US, US8228984[P]. 2012

[12] An Z Y, Liu P Q, Jiang H L. Dynamic Textures Retrieval Using the Integrated Wavelet-Based Surfacelet Transform[J]. Applied Mechanics & Materials, 2012, 263-266: 227-230

[13] 袁琴，吴宣够，熊焰. 小波树结构在贝叶斯压缩感知图像重构中的应用研究[J]. 计算机科学，2014，41(3)：314-318

[14] 汤敏，陈秀梅，陈峰. 基于 Contourlet 变换和 SPIHT 算法的彩色医学图像压缩[J]. 计算机科学，2014，41(1)：303-306

[15] Kaveh H, Moin M S, Razzazi F. A novel steganography approach for 3D polygonal meshes using Surfacelet Transform[C]// 2013 8th Iranian Conference on Machine Vision and Image Processing (MVIP). IEEE, 2013: 304-309